

PRUEBA JAVA SPRING BOOT – ANGULAR

La prueba tiene una duración de 8 horas, con una hora de almuerzo. Inicia a las 5:30 pm con un primer commit y se envía el link del repositorio respondiendo a este correo. Por favor realizar un commit hoy a las 11:59 pm para registrar los avances. El último commit se recepta al medio día de mañana.

BACKEND

Se requiere una API-REST para un Login que cumpla con las siguientes especificaciones:

1. **CRUD** básico de todas las tablas.

1.1. Para el CRUD de usuarios considerar lo siguiente:

Si se ingresa el usuario, el correo se generará en base a los datos del usuario que se está ingresando, de la siguiente manera:

Nombres: Juan Alberto

Apellidos: Piguave Loor

Identificación: 1206505024

Correo generado: jpiguavel@mail.com

(SI SE INGRESA A OTRO USUARIO Y SE OBTIENE UN MAIL DUPLICADO, DEBERÁ AGREGARSE UN NÚMERO, EJEMPLO: jpiguavel1@mail.com)

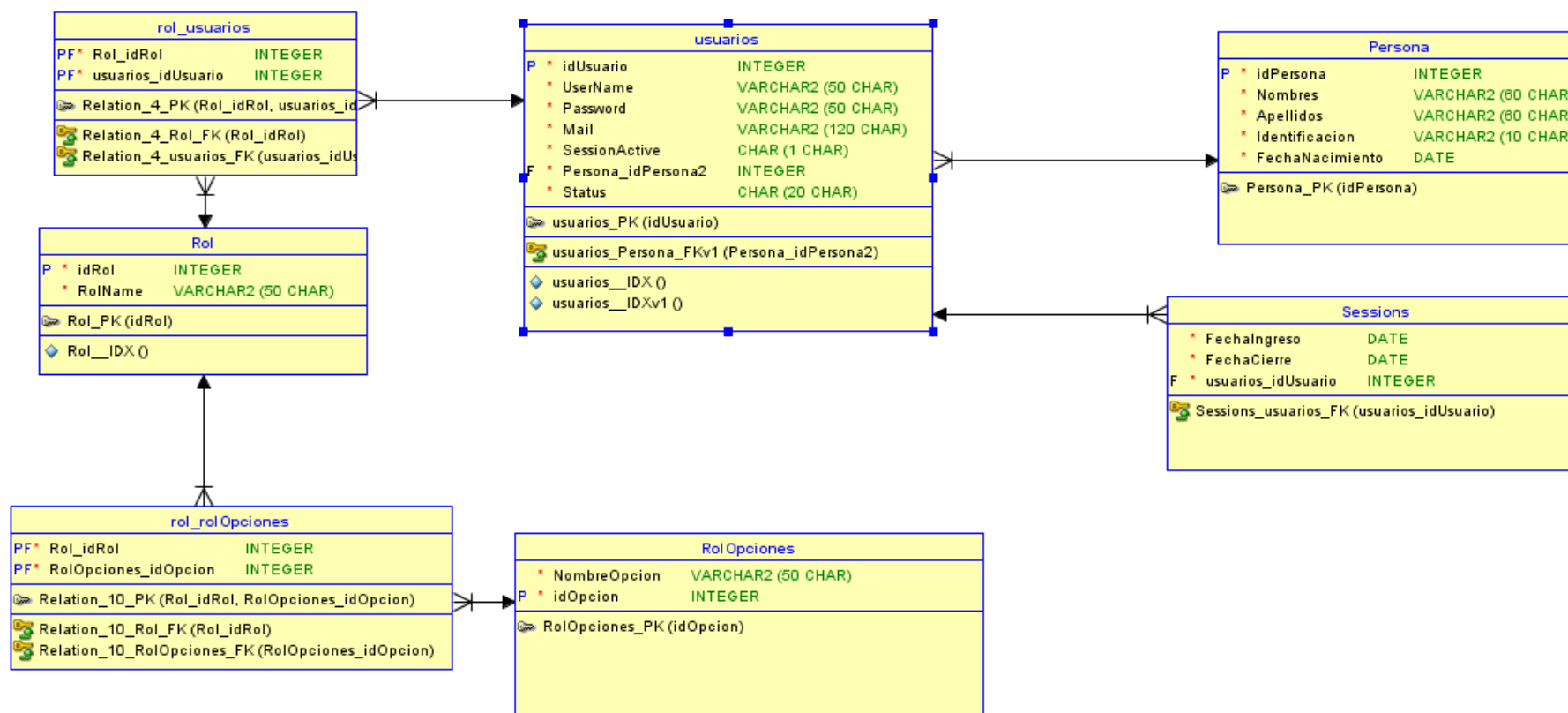
2. El nombre de usuario debe cumplir con los siguientes parámetros:
 - a. No contener signos
 - b. No debe estar duplicado
 - c. Al menos un número
 - d. Al menos una letra mayúscula
 - e. Longitud máxima de 20 dígitos y mínima de 8 dígitos
3. La contraseña debe cumplir con los siguientes parámetros:
 - a. Debe tener al menos 8 dígitos
 - b. Debe tener al menos una letra mayúscula
 - c. No debe contener espacios
 - d. Debe tener al menos un signo
4. La identificación debe cumplir con las siguientes especificaciones:
 - a. Debe tener 10 dígitos
 - b. Solo números
 - c. Validar que no tenga seguido 4 veces seguidas un número (Ejemplo: 100**8888**471)

Para el funcionamiento de la API considerar lo siguiente:

- I. El usuario solo puede tener 1 sesión activa. Se deben mostrar los mensajes correspondientes si hay una sesión activa.
- II. Se deben registrar los inicios y cierres de sesión.
- III. El inicio de sesión se puede realizar con correo y nombre de usuario.
- IV. El usuario tiene 3 oportunidades de ingresar la contraseña incorrecta. **Después** del tercer intento se bloquea al usuario y se debe retornar un mensaje indicando el bloqueo (Usar campo status en la tabla usuarios de la figura 1).
- V. Establecer roles para revisar el registro de inicios de sesión de un usuario específico.
- VI. Un usuario ingresado solo puede tener 1 cuenta registrada como máximo.

PRUEBA JAVA SPRING BOOT

Puede basarse en el siguiente diseño de base de datos:



Nota:

- Validar los datos de entrada con Json Schema
- Documentación básica de clases y métodos
- Mantener un desarrollo en capas
- Eliminaciones lógicas
- Implementar Swagger
- Usar JPA

Consideraciones:

- Para base de datos puede usar: SQL Server, PostgreSQL o MongoDB **(NO MYSQL)**.
- Puede usar Stored Procedures o funciones para el CRUD de Roles, registro de inicios de sesión, datos de persona, etc. **(Al menos una función o Stored Procedure en el caso de bases de datos relacionales)**
- Java 8, 11 o superior
- Buenas prácticas tanto en el código de la API.

FRONT END

Desarrollar las interfaces mínimas necesarias que permita realizar un login y posterior entrar el dashboard o página principal del aplicativo donde tendrá una opción de mantenimiento de persona.

- Interfaz Login
 - De permitir recuperar la contraseña
 - Validar que los datos sean ingresados antes de consumir el servicio
- Interfaz Mantenimiento de Usuario
 - De listar las personas activas e inactivas
 - Actualizar los campos de la persona, deberá considerar que campos son necesarios
 - Actualizar el estado

EL MENÚ DEBE CARGARSE DESDE LA BASE DE DATOS DE ACUERDO AL ROL

Consideraciones:

- Deberá realizar un desarrollo modularizado
- Las rutas deberán ser llamados a los módulos
- Protección de rutas
- Debe implementar un interceptor en todas las peticiones del aplicativo donde deberá adjuntar el jwt, cabeceras, etc. Datos necesarios para el consumo de los servicios (Opcional)
- Mostrar mensajes de error y exitoso
- Mostrar un loading o algo similar que le indique al usuario que el sistema está procesando su solicitud.
- Utilizar plantillas de diseño – El de su elección

Para la entrega:

- Adjuntar link de repositorio con la siguiente estructura de carpetas:
 - Backend
 - Frontend
 - Test (Scripts de bd con SP con datos de prueba y archivos Postman)
- Adjuntar script de base de datos
- Adjuntar archivo de postman con los CRUD de cada tabla y de las funciones del registro de usuarios y Login.